



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/628,960	07/28/2003	Carlos Bonilla	200309109-1	6175
22879 7590 02/25/2010 HEWLETT-PACKARD COMPANY Intellectual Property Administration 3404 E. Harmony Road Mail Stop 35 FORT COLLINS, CO 80528				
EXAMINER				
DAO, THUY CHAN				
ART UNIT		PAPER NUMBER		
2192				
NOTIFICATION DATE		DELIVERY MODE		
02/25/2010		ELECTRONIC		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM

ipa.mail@hp.com

laura.m.clark@hp.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte CARLOS BONILLA

Appeal 2009-003588
Application 10/628,960
Technology Center 2100

Decided: February 24, 2010

Before JEAN R. HOMERE, ST. JOHN COURTENAY III,
and STEPHEN C. SIU *Administrative Patent Judges*.

COURTENAY, *Administrative Patent Judge*.

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134(a) (2002) from the Examiner's rejection of claims 1-20. We have jurisdiction under 35 U.S.C. § 6(b).

We affirm in part.

STATEMENT OF THE CASE

INVENTION

The invention on appeal relates generally to emulation processing applications. More particularly, Appellant's invention is directed to emulation and native language interface testing. (Spec. 1).

ILLUSTRATIVE CLAIM

1. An emulation and native language interface test method comprising:
initializing an emulation language virtual machine;
wrapping native language code in a simulation test macro which creates simulated interfacing problems; and
examining reaction to said simulated interfacing problems when an emulation language application is run.

PRIOR ART

The Examiner relies upon the following reference as evidence:

Evans	US 6,826,746 B2	Nov. 30, 2004
-------	-----------------	---------------

THE REJECTION

The Examiner rejected claims 1-20 under 35 U.S.C. § 102(e) as anticipated by Evans.

GROUPING OF CLAIMS

Based on Appellant's arguments in the Briefs, we will decide the appeal on the basis of representative claims 1, 5, and 9.¹ See 37 C.F.R. § 41.37(c)(1)(vii).

ISSUES

Appellant argues specific limitations (App. Br. 18-24) that we address *infra*. The Examiner principally contends, *inter alia*, that Appellant is arguing limitations that are not claimed. (Ans. 8). The Examiner maintains that each claimed limitation is disclosed by Evans. (Ans. 8-16). Based upon our review of the administrative record, we have determined that the following issues are dispositive in this appeal:

Issue 1: Under § 102, has Appellant shown the Examiner erred by determining that the relied upon elements of Evans are arranged as claimed in independent claims 5 and 9?

Issue 2: Under § 102, has Appellant shown the Examiner erred in rejecting representative claim 1 by finding that Evans discloses or describes “wrapping native language code in a simulation test macro which creates simulated interfacing problems?”

¹ This Opinion considers only those arguments actually made. Arguments that Appellants could have made but chose not to make in the Briefs are waived. See 37 C.F.R. § 41.37(c)(1)(vii).

Issue 3: Under § 102, has Appellant shown the Examiner erred in rejecting representative claim 5 by finding that Evans discloses or describes “means for storing said information, including said instructions for testing said Java Native Language Interface, said means for storing said information coupled to said means for communicating information?”(underline in original) (*See* App. Br. 14-15).

Issue 4: Under § 102, has Appellant shown the Examiner erred in rejecting representative claim 9 by finding that Evans discloses or describes “running a Java application with simulated Java Native Language Interface problems if said Java Native Language Interface test mode is enabled?”

PRINCIPLES OF LAW

Anticipation under § 102

In rejecting claims under 35 U.S.C. § 102, “[a] single prior art reference that discloses, either expressly or inherently, each limitation of a claim invalidates that claim by anticipation.” *Perricone v. Medicis Pharm. Corp.*, 432 F.3d 1368, 1375 (Fed. Cir. 2005) (citing *Minn. Mining & Mfg. Co. v. Johnson & Johnson Orthopaedics, Inc.*, 976 F.2d 1559, 1565 (Fed. Cir. 1992)).

Anticipation of a patent claim requires a finding that the claim at issue ‘reads on’ a prior art reference. In other words, if granting patent protection on the disputed claim would allow the patentee to exclude the public from practicing the prior art, then that claim is anticipated, regardless of whether it also covers subject matter not in the prior art.

Atlas Powder Co. v. IRECO, Inc., 190 F.3d 1342, 1346 (Fed. Cir. 1999)
(citations omitted).

After the Examiner has presented evidence supporting a proffered prima facie case, Appellant has the burden of going forward to demonstrate error in the Examiner's position. *See In re Kahn*, 441 F.3d 977, 985-86 (Fed. Cir. 2006). Therefore, we look to Appellant's Briefs to show error in the Examiner's proffered prima facie case.

FINDINGS OF FACT

In our analysis *infra*, we rely on the following findings of fact (FF) that are supported by the record:

The Evans Reference

1. Evans discloses debugging a Java application that includes native method dynamic load libraries. (Col. 1, ll. 7-11).
2. Evans discloses an ICAT2 debugger probe run by the Java Virtual Machine which communicates to the separate instantiation of the JVM running the target application. Communications to the JVM are preferably TCP/IP messages. (Col. 5, ll. 7-12 and see col. 5, ll. 4-7).
3. Evans discloses that both the target application and the ICAT2 debugger are running on the same computer platform, so the communications between the various components of the system are usually internal to the computer itself. (Col. 5, ll. 13-17).
4. Evans discloses a Java Platform Debugger Architecture (JPDA) provides a Bootstrap object that allows access to the Java Debug Interface (JDI) interface. (Col. 5, ll. 53-55). Using the ICAT2 debugger probe (and JPDA) the application is launched when the probe function "launch_then_attach" is called. (Col. 5, ll. 64-66).

ANALYSIS

ISSUE 1

We decide the question of whether Appellant has shown the Examiner erred by determining that the relied upon elements of Evans are arranged as claimed in independent claims 5 and 9.

Regarding independent claim 5, we do not find persuasive Appellant's contention that the cited art is not arranged as claimed. (App. Br. 13-14). Appellant asserts that the portions of Evans relied on by the Examiner are "not part of a single system embodiment" and "do not appear in Evans as arranged in Claim 5 (e.g., coupled together to form a single system.)" (*Id.*). Regarding the claimed "means for communicating information," as recited in claim 5, we find Appellant points to four separate components (407, 453, 455, and/or 459) for support. (App. Br. 5, ¶3). It appears that "[i]nput component 453 [that] facilitates communication of information to computer system 450" (Spec. 16, 12-13) could be software, as contrasted with a defined structural component configured in a particular structural arrangement with other structural components. Evans discloses a CPU 151 (Fig. 15, col. 4, ll. 46-48) in a software development workstation 150 (Fig. 15) that we find inherently has storage means (i.e., a memory coupled to the CPU). Evans further discloses communication means (FF 2). Given the breadth of the claim term "coupled to" (i.e., any direct or indirect connection), we find Appellant's argument unavailing regarding the arrangement of claim 5.

Regarding independent claim 9, we also do not find Appellant's contention that the cited art is not arranged as claimed to be persuasive. (App. Br. 17-18). Appellant asserts that the parsed teachings of Evans relied

on by the Examiner are “not part of a single method embodiment” and “do not appear in Evans as arranged in Claim 9. (App. Br. 17). We find Appellant has not argued what the particular “arrangement” is regarding the software components of method claim 9, and how such purported “arrangement” differs from the relied upon portions of Evans, particularly given that there are no expressly claimed structural components in method claim 9.

ISSUE 2

We next decide the question as to whether Appellant has shown the Examiner erred by finding that Evans discloses or describes “wrapping native language code in a simulation test macro which creates simulated interfacing problems,” as recited in claim 1.

The Examiner contends that Evans recites a “simulation test macro” as a probe/launch method that creates simulated interfacing problems (Ans. 13). The Examiner further contends that Evans discloses the application under test includes both emulation language code (*e.g.*, Java) and native language code (*e.g.*, C/C++). (Ans. 12-13, *i.e.*, wrapping native language code in a simulation test macro). The Examiner also determined that Evans teaches that the probe function/launch method creates simulated interfacing problems. (*Id.*).

We find that Evans discusses debugging a Java application that includes native method dynamic load libraries. (FF 1). While it is reasonable to infer from Evans’ disclosure that the Java application contains or “wraps” the native language code, we nevertheless find the weight of the evidence supports Appellant’s arguments regarding the portions of Evans

relied upon by the Examiner as purportedly disclosing a “simulation test macro” that wraps the native language code. (*See* Reply Br. 3-4).

We find the Examiner has not sufficiently developed the record to clearly establish how either of Evans’ probe 41 or Evans’ launch method (col. 6, l. 4) fairly discloses the claimed “simulation test macro” that wraps the native language code. We agree with Appellant that probe 41 is shown as separate from the application in Figure 2 of Evans and also that Figure 6 of Evans shows various handlers of probe 41 that do not appear to contain or be wrapped around native language code. (App. Br. 4-5). We find that to sustain the Examiner based upon the cited portions of Evans would require speculation on our part. We decline to engage in speculation. Accordingly, we reverse the Examiner’s anticipation rejection of claim 1 and associated dependent claims 2-4.

ISSUE 3

We decide the question of whether under § 102, Appellant has shown the Examiner erred in rejecting representative claim 5 by finding that Evans discloses or describes “means for storing said information, including said instructions for testing said Java Native Language Interface, said means for storing said information coupled to said means for communicating information.” (underline in original).²

Appellant contends that no explanation was provided to highlight what is thought was relevant to the additionally cited section of Evans,

² We will direct our discussion only to the underlined portion of claim 5 as emphasized by Appellant. (*See* App. Br. 14-15).

namely col. 4, l. 65 – col. 5, l. 17 and col. 5 ll. 18-46. (Reply Br. 5). We disagree for the reasons discussed *infra*.

We find that Evans discloses that a communication means (TCP/IP connection) (FF 2), is coupled to (i.e., communicates with via direct or indirect connection) storage means (i.e., a computer inherently containing memory) which stores the target application. (FF 3). Thus, we find that Evans either expressly or inherently discloses corresponding structures that are capable of performing the claimed functions.

Therefore, we find Appellant has not sustained the requisite burden on appeal in providing arguments or evidence persuasive of error in the Examiner's anticipation rejection of representative claim 5, and claims 6-8 that fall therewith.

ISSUE 4

We decide the question of whether Appellant has shown under § 102, the Examiner erred in finding that Evans discloses or describes “running a Java application with simulated Java Native Language Interface problems if said Java Native Language Interface test mode is enabled.” (Representative claim 9).

Appellant contends that simultaneous debugging of Java code and C/C++ code is not the same as and does not teach, “‘running a Java application with simulated Java Native Language Interface problems if said Java Native Language Interface test mode is enabled’, as recited in claim 9.” (App. Br. 19)(quote in original).

We disagree. In particular, we find Appellant has not argued a particular meaning or definition that defines the metes and bounds of the claimed “simulated Java Native Language Interface problems.” (Claim 9).

Given this lack of definition, we find the weight of the evidence supports the Examiner's findings with respect to the argued limitation of claim 9. (Ans. 20). More particularly, we find that the application is launched when the probe function "'launch_then_attach' is called" (enabled), passing arguments to each connector. (Col. 5, ll. 64-66; *see also* FF 4). We further note that Appellant had an opportunity to dispute the Examiner's findings with respect to the aforementioned limitation, and did not do so in the Reply Brief (aside from restating the arrangement argument) (*See* Reply Br. 7, last paragraph).

Therefore, we find Appellant has not sustained the requisite burden on appeal in providing arguments or evidence persuasive of error in the Examiner's anticipation rejection of representative claim 9 and claims 10-20 that fall therewith.

CONCLUSIONS

Based on the findings of facts and analysis above:

Under § 102, Appellant has not shown the Examiner erred by determining that the relied upon elements of Evans are arranged as claimed in independent claims 5 and 9.

Under § 102, Appellant has shown the Examiner erred in finding that Evans discloses or describes "wrapping native language code in a simulation test macro which creates simulated interfacing problems." (Claim 1). Thus, Appellant has also shown that the relied upon elements of Evans are not arranged as claimed in independent claim 1.

Under § 102, Appellant has not shown the Examiner erred in finding that Evans discloses or describes “means for storing said information, including said instructions for testing said Java Native Language Interface, said means for storing said information coupled to said means for communicating information.” (Claim 5).

Under § 102, Appellant has not shown the Examiner erred in rejecting representative claim 9 by finding that Evans discloses or describes “running a Java application with simulated Java Native Language Interface problems if said Java Native Language Interface test mode is enabled.” (Claim 9).

ORDER

We reverse the Examiner's rejection of claims 1-4 under 35 U.S.C. § 102(e).

We affirm the Examiner's rejection of claims 5-20 under 35 U.S.C. § 102(e).

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED-IN-PART

pgc

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
3404 E. Harmony Road
Mail Stop 35
FORT COLLINS, CO 80528